Eclipse Plug-in Development Research Project

Report 16

UNIVERSITY AT BUFFALO

November 11, 2015

Authored by: Chern Yee Chua

Eclipse Plug-in Development Research Project

Report 16

Short report

Couple of things:

1. Data Recording mechanism
   By manipulating threads, now the data recording will not take place when there is short pause while typing code (I set it less than 2 seconds). If the pause is more than 2 seconds, then it will record the data.

2. Recorded data
   When you add/remove a new line, it will show what type of node it is. If it is not a node, it probably is an error. Hence, error message at that point will be shown. For example,

```
 0010 int a;
+0011 int b; [Field Declaration]
 0012 int c;
```

```
 0010 int a;
+0011 int b [Syntax error, insert ";" to complete FieldDeclaration]
 0012 int c;
```

   Note: those messages are retrieved from eclipse API, not something that I made up ☺

3. Different cases
   - One line added
     For example,

```
 0010 int a;
+0011 int b; [Field Declaration]
 0012 int c;
```

     For the remove case, I do not include the information at the end of the line, the reason is that before removing a line, there should exist a record showing the added line with the node information at the end of each specific line, and that already tells what type the element is. (Difficulty in matching the line number with the removed lines in previous compilation unit)

- One line changed
  For example,

```
  0010 int a;
+0011 int b; [Field Declaration]
 -0011 int b
  0012 int c;
```

- Multiple lines added
  For example,

```
+0011 int b; [Field Declaration]
+0012 int c; [Field Declaration]
+0013 int d; [Field Declaration]
```

  Note: for multiple lines (more than 1), I did not include the previous and next line of the added lines due to the complexity of the algorithm.

4. Printing source file
   Printing source file is no longer based on the time interval, but rather based on the number of character difference. If the number of character difference is larger than a certain number (1000 characters for example), then the source of current compilation unit will be generated. I added another column for line difference between two subsequent recorded source files. So there is like two kinds of data recording mechanism, one is detected whenever there is a change in eclipse, the other one is detected when the changes in number of characters in the file larger than a specific number.

5. Issues
   There is an issue with the error case. When there's error, the information at the end of line will not be shown sometimes. For example, you added this line

```
+0020 int a;
```

   And the node description will sometimes be omitted. The reason is that there probably exists error before or after this line of code, and that the root ASTNode traversal couldn't detect all the nodes properly (ASTNode is like an enigma). One thing that is guaranteed will be there is the added/removed lines with line numbers.

Similarly, in some case,

```
+0024 int a
```

will not have error message (missing ‘;’) because of some other errors that already exist in the editor. But generally, if the code is written properly, as in if there’s error in the line, fix it right away, the data output will be much accurate.

In a nutshell, this plugin update is much cleaner and simpler comparing to the previous implementation. It’s also way more effective and easier to view the data. Those issues mentioned above is not really something really important. To prevent SQL injection, I replace all the single quote with “`” as a temporary solution. That is, when student type “’” this, it will be “`” instead of “’”, yeah, I know, just little thing that I want to show ☺

I guess that’s it. Thanks in advance!